Rex Holmes F39-W7-Assessment

Array	Insert Time	Append Time	Total Time
extraLargeArray	696.482916 ms	4.988667 ms	701.471583 ms
largeArray	5.795959 ms	334 μs	6.130959 ms
mediumArray	121.5 μs	75 μs	196.5 μs
smallArray	21.25 μs	42.25 μs	63.5 μs
tinyArray	12.75 μs	36.666 μs	49.416 μs

I see that as the arrays get smaller, obviously, so does both the insert and append time, but eventually the insert time ends up taking less time than the append time the smaller the array gets. The doublerAppend function has a time complexity of O(n). The doublerInsert function has a time complexity of O(n^2). The doublerAppend function actually scales better though because the time it takes to execute the function grows linearly with the size of the input array, compared to the doublerInserter function which grows quadratically with the size of the input array. This is because the unshift() is inside the for loop and, for each iteration, it has to shift all the elements that were previously inserted. Like we were taught in the lectures, it would be better to just build the array in the reverse order and then reverse it at the end and make the time complexity O(n).